

TensorFit: A torch-based tool for ultrafast metabolite fitting of large MRSI data sets

Federico Turco^{1,2}  | Milena Capigliani^{1,2}  | Guodong Weng¹  | Johannes Slotboom¹ 

¹Support Center for Advanced Neuroimaging, Institute for Diagnostic and Interventional Neuroradiology, University of Bern, Bern, Switzerland

²Graduate School for Cellular and Biomedical Sciences, University of Bern, Bern, Switzerland

Correspondence

Federico Turco, Support Center for Advanced Neuroimaging, Institute of Diagnostic and interventional Neuroradiology, Inselspital, Freiburgstrasse 18, 3010, Bern, Switzerland.

Email: federico.turco@extern.insel.ch

Funding information

European Union's Horizon 2020 research and innovation programme, Grant/Award Number: 813120; Schweizerische Nationalfonds zur Förderung der wissenschaftlichen Forschung, Grant/Award Numbers: 182569, 207997

Abstract

Purpose: To introduce a tool (TensorFit) for ultrafast and robust metabolite fitting of MRSI data based on Torch's auto-differentiation and optimization framework.

Methods: *TensorFit* was implemented in *Python* based on Torch's auto-differentiation to fit individual metabolites in MRS spectra. The underlying time domain and/or frequency domain fitting model is based on a linear combination of metabolite spectroscopic response. The computational time efficiency and accuracy of *TensorFit* were tested on simulated and in vivo MRS data and compared against TDFDFit and QUEST.

Results: *TensorFit* demonstrates a significant improvement in computation speed, achieving a 165-times acceleration compared with TDFDFit and 115 times against QUEST. *TensorFit* showed smaller percentual errors on simulated data compared with TDFDFit and QUEST. When tested on in vivo data, it performed similarly to TDFDFit with a 2% better fit in terms of mean squared error while obtaining a 169-fold speedup.

Conclusion: *TensorFit* enables fast and robust metabolite fitting in large MRSI data sets compared with conventional metabolite fitting methods. This tool could boost the clinical applicability of large 3D MRSI by enabling the fitting of large MRSI data sets within computation times acceptable in a clinical environment.

KEYWORDS

3D MRSI, deep learning frameworks, GPU optimization, metabolite fitting, torch auto-differentiation

1 | INTRODUCTION

Alterations in the chemical concentrations of specific metabolites in the brain are often associated with certain pathologies. MRSI enables noninvasive quantification of metabolites and has applications in tumor

classification,^{1–4} treatment follow-up,^{5–7} and the study of metabolic disorders leading to neurological diseases.⁸ However, low resolution, spectral quality, and difficulties in interpretation make the clinical application of MRSI challenging.⁹

This is an open access article under the terms of the [Creative Commons Attribution](https://creativecommons.org/licenses/by/4.0/) License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2024 The Authors. *Magnetic Resonance in Medicine* published by Wiley Periodicals LLC on behalf of International Society for Magnetic Resonance in Medicine.

Substantial research has been undertaken to solve these factors and augment the efficacy of MRS, primarily through advancements in pulse sequences and artifact removal methods.^{10–12} One promising approach is using a 3D echo-planar spectroscopy imaging (EPSI) sequence,^{13–15} which improves spatial resolution and reduces the acquisition time, obtaining information of larger volumes by simultaneously acquiring multiple spectra. However, an increase in the amount of acquired spectra results in longer processing times. Whole-brain acquisitions can yield up to 131 072 spectra for a $64 \times 64 \times 32$ 3D-MRSI data set, as is the case of ECCENTRIC,¹⁶ resulting in metabolite fitting times taking up to hours to compute on dedicated servers.

There are several implementations available for fitting of MRS data, such as LCMoDel,¹⁷ TARQUIN,¹⁸ QUEST,¹⁹ and TDFDFit,²⁰ each differing in fitting domain, minimization algorithm, and spectral models used. Existing software implementations of the mentioned methods lack GPU use, and some lack even CPU parallelization. Although this might not be a major issue for sequences involving only a small number of spectra, such as single-voxel spectroscopy or 2D MRSI, it is a big issue when analyzing data from high-resolution MRS methods, like EPSI. This is particularly problematic in a clinical setting where timely results are essential. Therefore, new time-efficient fitting tools are needed to accommodate the requirements of large data-acquisition methods.

A prior study explored the use of GPUs for rapid metabolite fitting.²¹ However, this implementation was limited to metabolites with Gaussian lineshapes and lacked the incorporation of prior knowledge and the use of simulated metabolite bases. Simultaneously, research has focused on developing deep learning (DL) techniques applied to metabolite quantification.^{22–26} These DL methods, although fast, have the drawback of being biased by the data set that is used for training.²⁷ Furthermore, DL methods have difficulties to generalize across quantification models, sequences, and hardware specifications (e.g., field strength, scanner manufacturer, T_E , T_R).

Despite all these challenges, the principles and efficient tools developed for error minimization in the field of DL can be applied for efficient curve fitting. Here, we propose using auto-differentiation tools from DL frameworks (Torch²⁸)—not to train a neural network for generalizing across multiple spectra, but to perform highly efficient fitting on every individual spectrum. Although similar concepts have been recently applied to curve fitting in other fields,^{29–31} to the best of our knowledge, this approach has not been previously applied for metabolite fitting in MRSI.

In this work, we develop an ultrafast tool based on a linear combination of baseline metabolites. We assess its precision and time efficiency compared with other linear

combination methods, such as QUEST and TDFDFit, on simulated and in vivo data. This work lays the foundation for faster spectral fitting implementations that will boost the use of large data-size MRSI recordings in a clinical setting.

2 | METHODS

2.1 | Fitting model

The general model to represent the time response of the spectroscopy signal $s(t)$ can be written as

$$s(t) = \rho(t) + e(t) + \mu(t) \quad (1)$$

where $\rho(t)$ is a linear combination of each metabolite response in time domain (TD); $e(t)$ represents Gaussian noise over the signal; and $\mu(t)$ corresponds to the macro-molecular baseline. The metabolites response $\rho(t)$ is defined as a linear combination of Voigt lineshapes,²⁰ such as:

$$\rho(t) = \sum_m^M A_m B_m(t) e^{\left(-\frac{t}{T_{2,m}} - \frac{t^2}{T_{G,m}} + i(\omega_m t + \phi_m)\right)} \quad (2)$$

where A_m , $T_{2,m}$, $T_{G,m}$, ω_m , and ϕ_m represent the area, Lorentzian damping, Gaussian damping, frequency shift, and zero-order phase for each metabolite m , respectively. The value of $B_m(t)$ is the amplitude-normalized quantum mechanical simulated time response, including relaxation and J-modulation effects. The simulation corresponds to the used acquisition sequence and particular metabolite characteristics, and the time t is a discrete vector with the N sampling times.

The Cramér-Rao Lower Bound (CRLB) defines the lowest theoretical error in an unbiased parameter estimation. It has been proven to minimize the number of free parameters in the model reduces errors.^{32,33} This can be accomplished by incorporating prior knowledge, which means linking the values of parameters to estimate as follows^{20,34,35}:

$$\omega_m = \Delta\omega_m + \omega_c \quad (3a)$$

$$T_{2,m}^{-1} = \Delta T_{2,m}^{-1} + T_{2,c}^{-1} \quad (3b)$$

$$T_{G,m}^{-1} = T_{G,c}^{-1} \quad (3c)$$

$$\phi_m = \Delta\phi_m + \phi_c \quad (3d)$$

where ω_c , $T_{2,c}^{-1}$, $T_{G,c}^{-1}$, and ϕ_c are common free parameters, as $B_m(t)$ handles individual differences in T_2 , T_G , ϕ , and ω .

On the other hand, $\Delta\omega_m$, $\Delta T_{2,m}^{-1}$, and $\Delta\phi_m$ are values that define the prior knowledge and remain constant during the model fitting process. These parameters are necessary for fine-tuning the simulation basis set. Applying the relations defined in Eq. (3) to Eq. (2), we obtain

$$s(t) = e^{-\left(T_{2,c}^{-1} + T_{2,m}^{-1}\right)t + i\left(2\pi\omega_c t + \frac{\pi\phi_c}{180}\right)} \times \sum_m^N A_m B_m(t) e^{-\Delta T_{2,m}^{-1} t + i\left(2\pi\Delta\omega_m t + \frac{\pi\Delta\phi_m}{180}\right)} \quad (4)$$

The incorporation of prior knowledge not only reduces the minimal errors in the parameters to estimate but also increases the tool's robustness and computational speed. In this work, prior knowledge is defined using the spectrIm-QRMS³⁶ spectra modeling functionality, which has a graphical user interface for setting the model and the relationship between metabolites (available at <https://spectrim.diskstation.me/spectrImWeb>). Using the same graphical interface, the user defines the initial amplitude for each metabolite. The fitting model and initial values were defined for an average spectrum on a clinical data set of 256 spectra obtained with the same sequence type and acquisition protocol. This approach guarantees a good estimate of the starting parameter. These starting values were used in all compared fitting methods tested. Additional details of both prior knowledge models applied are presented in Table S1 and S2.

The fitting in TensorFit can be performed in frequency domain (FD) as well as in time domain (TD) by computing the fast Fourier transform (FFT) of $s(t)$, as follows:

$$S(\omega) = \text{FFT}(s(t)) \quad (5)$$

Application of nonlinear least squares fitting is the expert consensus³⁷ for metabolite fitting. It involves searching for parameters that minimize the cost function L , as follows:

$$\underset{\theta}{\text{argmin}} L(S(\omega), y(\omega)) \quad (6)$$

where the vector θ are the free parameters of the model, and $S(\omega)$ and $y(\omega)$ are the modeled and measured spectrum, respectively.

State-of-the-art fitting software uses different optimization options for this problem. FitAid,³⁴ QUEST, and LCModel use the Levenberg-Marquardt³⁸ algorithm as their base optimization method. ProFit³⁵ and TARQUIN, on the other hand, use VARPRO³⁹ as their base algorithm for fitting. Finally, TDFDFit²⁰ applies conjugate gradient descent⁴⁰ to fit the spectra. The implementation presented in this work will be compared against TDFDFit and QUEST. We used QUEST because of its performance on the ISMRM fitting challenge 2016,⁴¹ and TDFDFit because

it was available, in-house-developed, and a commonly used algorithm.

2.2 | TensorFit implementation

TensorFit was implemented using the Torch v1.13.1 framework in Python v3.10.9. Equations (4) and (5) were implemented as a Torch computational graph as shown in Figure 1, where the free parameters (A_m , ω_c , ϕ_c , $\frac{1}{T_{2,c}}$, and $\frac{1}{T_{2,m}}$ in Eq. [4]) are trainable tensors defined using "torch.nn.parameters.Parameter." These tensors are forward-propagated through the computational graph to compute the model response. The error between the model response and the target spectra is backpropagated through the network to update the parameters. After this iteration, the updated parameters of the computational graph will be closer to those that minimize the error. After several iterations, the error between the response and the target spectra will reach a minimum.

We chose Torch, as this framework allows us to

- Define complex formulas as a nn.Module and take full advantage of automatic differentiation;
- Study a wide range of minimization algorithms (optimizers), including first-order and second-order minimization and several strategies to avoid ending up in local minima;
- Redefine or modify the computational graph during the process of fitting;
- Use efficient and optimized parallel GPU operations, making it suitable for high-dimensionality problems; and
- Use complex numbers in all the needed operations, optimizers, and backpropagation algorithms.

To evaluate the fit quality between the (measured) spectrum to be fitted $y(k, x)$ and the modeled spectrum $S(k, x)$, with k representing the index of the spectrum in the batch, we used a loss function defined as

$$L(S', y') = \frac{\sum_k \sum_x \left| S'(k, x) - y'(k, x) \right|^2}{N_x N_k} \quad (7)$$

where $S'(k, x)$ and $y'(k, x)$ are the spectral amplitude offset removed versions of $S(k, x)$ and $y(k, x)$. The values of N_x and N_k are the number of points in the signal and the number of spectra, respectively. Here, x represents either the time t in TD or ω in FD; both variables are limited to user-selected ranges, to focus the fitting on a particular spectral region. In the underlying study, minimization was performed in FD, and the full spectral range was used.

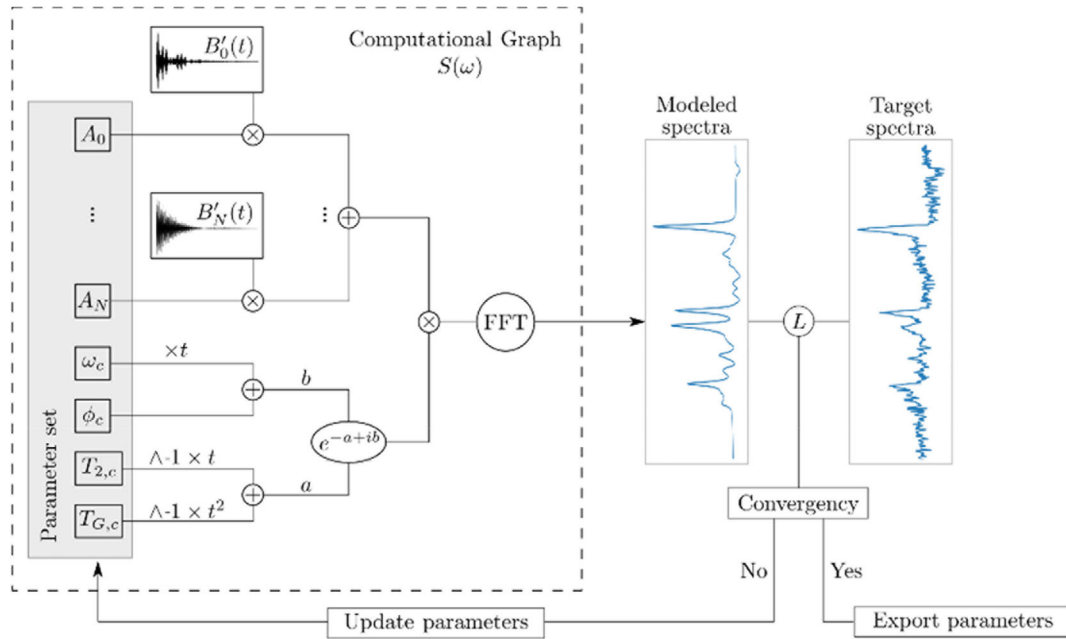


FIGURE 1 TensorFit implementation diagram. The tensor computational graph calculates the spectral response in frequency domain. The mean squared loss is calculated between the modeled and the target (simulated or measured) spectra. When the convergence is not satisfied, the parameters are updated, and the process is repeated until convergence is reached. FFT, fast Fourier transform.

Spectral offset removal was performed by spectrum-wise subtraction of the average between the first 200 points in FD.

The fitting consists of two phases. In Phase 1, we iterate fitting a truncated model, where the simulated basis set $B_m(t)$ is truncated to contain only the first quarter of the points. This way, the number of complex exponentials and matrix multiplications is drastically reduced, improving computation speed. During this phase, the iteration ends when the loss function $L(S', y')$ improves less than $\Delta L(\%) = 0.01\%$ over the last N iterations. In Phase 2, we iterate to fit the full model (i.e., without truncation). The criteria to terminate this final iteration is the same as before. In this work, the value of N is 20. Both $\Delta L(\%) = 0.01\%$ and $N = 20$ were a good compromise between time and accuracy on the simulated data set.

Once convergence is reached, the CRLB is computed for $\theta = \theta_{min}$ as

$$CRLB_m = \sqrt{\left(\frac{\Re e(D^H D)}{\sigma^2} \right)^{-1}_{m,m}} \quad (8)$$

where $D_{i,j} = \frac{\partial S(x_i)}{\partial \theta_j}$ is the derivative for each frequency with respect to each free parameter, and σ^2 is the variance of the Gaussian noise. The derivative is extracted directly from the computational graph after the last iteration. During minimization, hard constraints were implemented as part of the computational graph using

the function “torch. clamp”. This is equivalent to assigning $\theta = \min(\max(\theta_{input}, l_b), u_b)$, where l_b and u_b are the lower and upper bound for a certain parameter θ . The corresponding gradient is set to zero when $\theta_{input} < l_b$ or $\theta_{input} > u_b$. This allows the free parameters to move between the limits but never go further.

We used different optimizers provided by the Torch library for loss minimization. In this work, we evaluated the performance of Rprop,⁴² Adam,⁴³ and stochastic gradient descent (GD) as first-order methods, and AdaHessian⁴⁴ and limited-memory broyden-fletcher-goldfarb-shanno (LBFGS)⁴⁵ as second-order methods.

2.3 | Simulated data set

We simulated a spectra data set to evaluate the accuracy and time efficiency of each studied tool, as this is not feasible in clinical cases due to the absence of ground truth. The model consists of nine metabolites: choline (Cho), glutamate, lactate, glutamine (Gln), creatine (Cr), aspartate, N-acetylaspartate chemical group $^2\text{CH}_3$ (NAA), myo-inositol, and N-acetylaspartate chemical group $^3\text{CH}_2$. Each basis was simulated using NMRScopeB⁴⁶ for the 3T scanner with a semiLASER 2D-MRSI sequence, TE = 135 ms, TR = 1500 ms, and 1024 points. The Gaussian width ($\frac{1}{T_{G,c}}$) was set to zero, to assume only Lorentzian lines and maintain comparability with QUEST, which does not fit Gaussian lineshapes.

To ensure that our simulated data set included the range of spectral shapes and noise levels typically observed in clinical data, we randomly sampled from uniform distributions in the indicated value ranges, as presented in Table 1. The area range was proportional to its corresponding starting value, whereas the frequency shift range was relative to its initial value in the model. More details of both models used in this work (3 T and 7 T) can be found in Tables S1 and S2, respectively. Additionally, we varied the SNR for each simulated spectrum to assess the performance of various analysis techniques across a range of SNR between 5 and 50, levels commonly encountered in in vivo spectroscopy. The SNR was calculated as the maximum spectral amplitude divided by the SD of the noise in the spectral baseline from the first 200 points in FD as follows:

$$SNR = \frac{\max(S(\omega))}{std(S(\omega[0,200]))} \quad (9)$$

We calculated the difference in frequency shift between the initial model spectrum and the measured or simulated spectrum to improve the initial seed values and increase the probability of finding an acceptable minimum. This was achieved by identifying the ω_c value that maximizes the cross-correlation between the two signals. This step increases the method's robustness and lowers the number of outliers by obtaining a better seed for the initial frequency shift.

2.4 | In vivo data set

For testing on in vivo data, we use a data set obtained from scans conducted on a 7T scanner (Terra; Siemens Healthineers, Erlangen, Germany) using a spectral-editing sequence named SLOW-editing¹⁵ with TE = 68 ms, TR = 1500 ms, and FOV = 280 × 100 × 70 mm. One data set typically consists of a matrix with 65 × 42 × 10 = 27 300 spectra, from which 20 202 were discarded for being outside the brain region or voxels near the skull due to too strong lipid contamination. On the remaining 7098 voxels, B1⁻/B1⁺ correction was performed using water reference data (i.e.,

Scorr = $\frac{S}{\text{WaterArea}}$). For a detailed description of the model used to fit this data set, see Table S2.

Like the approach used on these described simulated data, the maximum cross-correlation was determined to correct the frequency shift between the model and measured spectrum. Before fitting, the spectral amplitude offset was also corrected in FD, determined as the mean on the downfield part of the spectra (precisely, the leftmost 200 points).

2.5 | Performance analysis

2.5.1 | Performance of optimizers

To define the best-performing Torch optimizer for the complexity of our problem, we performed metabolite fitting on identical simulated data for Rprop, Adam, GD, Ada-Hessian, and LBFGS. We used a subset of the simulated data set containing 2048 spectra. For each optimizer, we first searched the optimal learning rate lr , which resulted in $lr = 10$ for GD and $lr = 1$ for the other four optimizers. Considering the optimal lr of each optimizer, we computed the percentual error as a function of the number of iterations as $\frac{100(A_m - A'_m)}{A'_m}$, where A'_m is the ground truth for each of the most abundant metabolites (Cr, Cho, NAA, and Gln). The number of iterations was considered in the range of 0 to 50.

2.5.2 | TensorFit validation

We evaluated the performance in accuracy and time efficiency of TensorFit in comparison with established fitting methods such as TDFDFit and QUEST. First, we analyzed the percentual error for a simulated data set containing 32 768 spectra each, with SNR varying between 5 and 50. These metabolites were fitted using the same 3T model used for the simulation. The percentual error was averaged between all the spectra in the data set. When compared with QUEST, the function used to fit in TensorFit is Eq. (4) in FD but removing the free parameter ω_c and considering

TABLE 1 Parameter ranges in which values were randomly sampled. The area range is defined respective to the initial area of each metabolite defined in the model (for more details, see Table S1).

Parameter	Minimum	Maximum
Area A_m (a.u.)	0.1×	2×
Frequency shift ω_c (Hz)	−20	20
Lorentzian width (Hz)	1.5	6
Phase ϕ_c (°)	−45	45

that each metabolite has its own ω_m , which varies within a predefined $\Delta\omega_m$. This adjustment is necessary due to the specific model definition within QUEST. In addition, QUEST does not allow Gaussian decay. Therefore, the free parameter $\frac{1}{T_{G,c}}$ is removed from the fitting model. This use case is included in the TensorFit implementation, but for clarity, we refer to it as TensorFit + MetShift. The Gaussian decay was also removed from the TDFDFit model. In this case, both methods had the same constraints, which are ± 20 Hz for each metabolite frequency shift and $\pm 180^\circ$ for the global zero order.

The execution time was measured as the time necessary to fit a certain number of spectra, excluding the loading time of each algorithm. Fitting with QUEST was performed within a jMRUI⁴⁷ pipeline. Because of the memory limitation of the 32-bit JVM (Java Virtual Machine), the maximum number of spectra that we could fit in one batch using jMRUI was 4096. When using TDFDFit, a parallelized call was implemented using *Python*. For TensorFit, the fitting was performed directly in a batch of the needed amount of spectra. The total time required for each fitting was determined for a data set of varying sizes between 32 and 32 768 for TensorFit and TDFDFit and ranging from 32 to 4096 for QUEST and TensorFit + MetShift. All presented CPU results were obtained on an AMD Ryzen (72 700 \times , 8-core, 3.7 GHz, 32 GB RAM), whereas the GPU performance was measured on a Nvidia GeForce GTX 1060 (6 GB, 1280 CUDA-cores).

2.5.3 | In vivo application

For the in vivo spectral editing case, the comparison was performed only between TensorFit and TDFDFit, given the fact that QUEST can only handle Lorentzian line-shapes. The whole brain was fitted and directly compared between both methods, with metabolite maps generated using SpectrIm-QRMS. For comparison, metabolite maps for gamma-aminobutyric acid (GABA), glutamate + glutamine, and NAA are shown for both methods. CRLB was computed and averaged among all the voxels shown. To numerically evaluate the performance of both methods, we calculated the ratio between the TensorFit and TDFDFit loss. This is represented as a histogram showing the distribution across the 7098 spectra.

3 | RESULTS

3.1 | Optimizer evaluation

To determine the optimal Torch optimizer, we fitted the 3T prior-knowledge model for five different optimizers for

the same 2048 simulated spectra. Figure 2A–D shows the percentual error for the metabolites NAA, Cr, Cho, and Gln, respectively, for each optimizer as a function of the number of iterations. Similar behavior is observed for the four metabolites. Rprop converges in fewer iterations than AdaHessian, Adam, and GD, while needing more iteration than LBFGS. Nevertheless, LBFGS evaluates the loss function several times in each iteration, causing the effective convergence time to be higher than that of Rprop. The label in Figure 2A shows each optimizer's total computation time to reach iteration number 50. It is noticeable that, although LBFGS converges in fewer iterations, the convergence time is more than 10 times larger than that of Rprop. Given the faster convergence speed, Rprop was adopted for all subsequent analyses with TensorFit.

3.2 | Precision and time efficiency

For evaluating the precision of TensorFit, we fitted the 3T model and evaluated the percentual error against the ground truth used for the corresponding simulation. Figure 3 shows the percentual error averaged among all spectra for TensorFit and TDFDFit in blue and light red, respectively. The solid red bar corresponds to the error for TDFDFit when removing outliers, defined as those cases in which the common frequency shift (ω_c) differs from the ground truth more than 20 Hz, represented as “Filtered TDFDFit.” Those cases represent 4% of the data set. The percentual error for TensorFit + MetShift and QUEST is shown in yellow and green, respectively. As there is no common frequency shift to consider, the outliers were identified as those cases in which the most abundant metabolites (NAA, Cho, and Cr) had errors exceeding four SDs from the ground truth, represented as “Filtered QUEST.” This accounted for nearly 5% of the data set. In the case of TensorFit, no outliers were found. Additionally, we present the average CRLB computed using the ground-truth parameters. An extra comparison between TensorFit and TDFDFit, including Gaussian decay, is shown in Figure S1.

To evaluate whether the method can be used for processing large data sets, we fitted the simulated data and measured the execution time. Figure 4 shows the execution time for each method as a function of the number of spectra. Figure 4A presents the execution time when comparing TensorFit with TDFDFit. The execution time increases linearly for TDFDFit, whereas TensorFit on the CPU follows a similar rate with a speed-up of 11 \times . When running on the GPU, TensorFit remains strongly dominated by the Torch's GPU context initialization, starting to rise when reaching a data-set size of 2048, with a maximum speed-up of 165 \times for 32 k spectra. Figure 4B

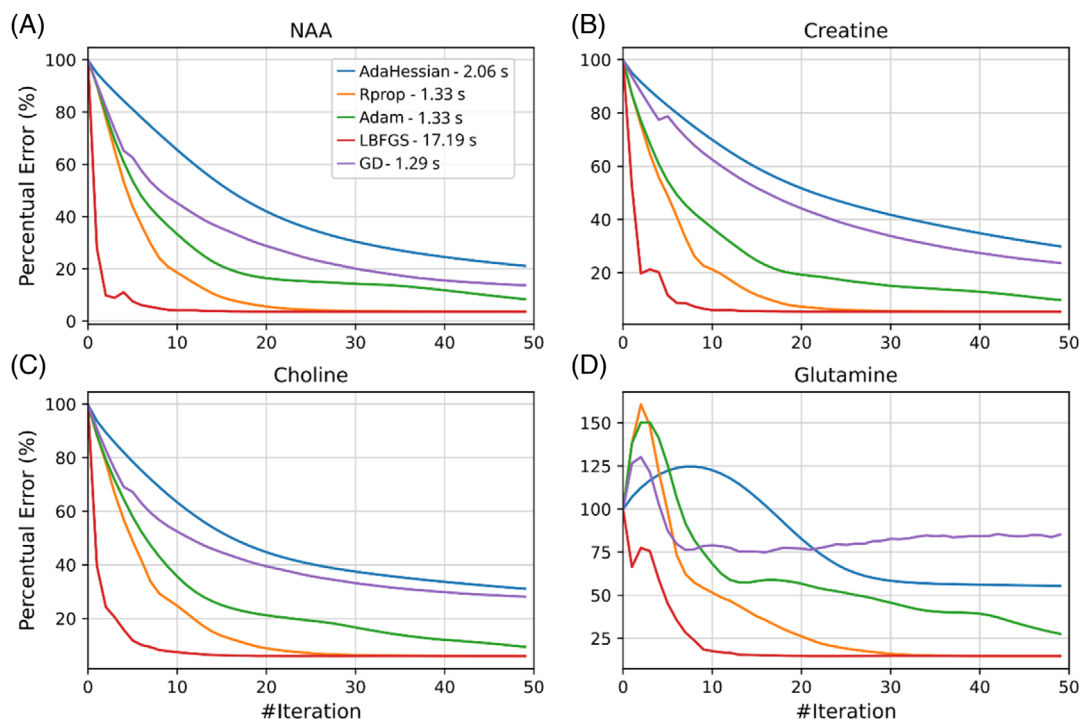


FIGURE 2 Convergence comparison of TensorFit optimizers. The percentage error (%) is plotted against the iteration number for AdaHessian, Rprop, Adam, limited-memory broyden-fletcher-goldfarb-shanno (LBFGS), and gradient descent (GD). Results are shown for N-acetylaspartate chemical group $^2\text{CH}_3$ (NAA; A), creatine (B), choline (C), and glutamine (D). The label in (A) states each optimizer's total computation time.

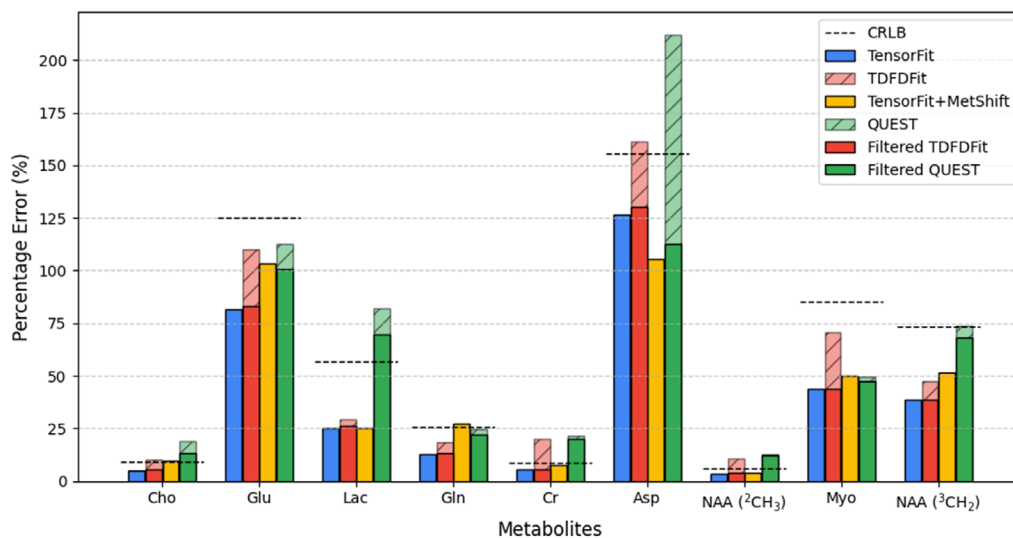


FIGURE 3 Precision analysis for all fitting tools used. The percentage error (%) with respect to the ground truth is displayed for each of the nine metabolites considered in the 3T prior-knowledge model. For TDFDFit and QUEST, the mean error without outliers is shown with solid bars (i.e., “Filtered TDFDFit” and “Filtered QUEST”). The black dotted line represents the averaged Cramér-Rao Lower Bound (CRLB) obtained using the ground truth. The metabolites labels, with no chemical group information, contain all of them. Asp, aspartate; Cho, choline; Cr, creatine; Gln, glutamine; Glu, glutamate; Lac, lactase; Myo; myo-inositol; NAA, N-acetylaspartate.

illustrates the execution time for QUEST and TensorFit + MetShift. In this case, we observe the same behavior as before, but with a smaller speed-up, 7 times faster on the CPU, and 115 times compared with QUEST.

3.3 | In vivo performance

After validating the method with simulated spectra, we tested its performance on in vivo spectra. Figure 5A–C

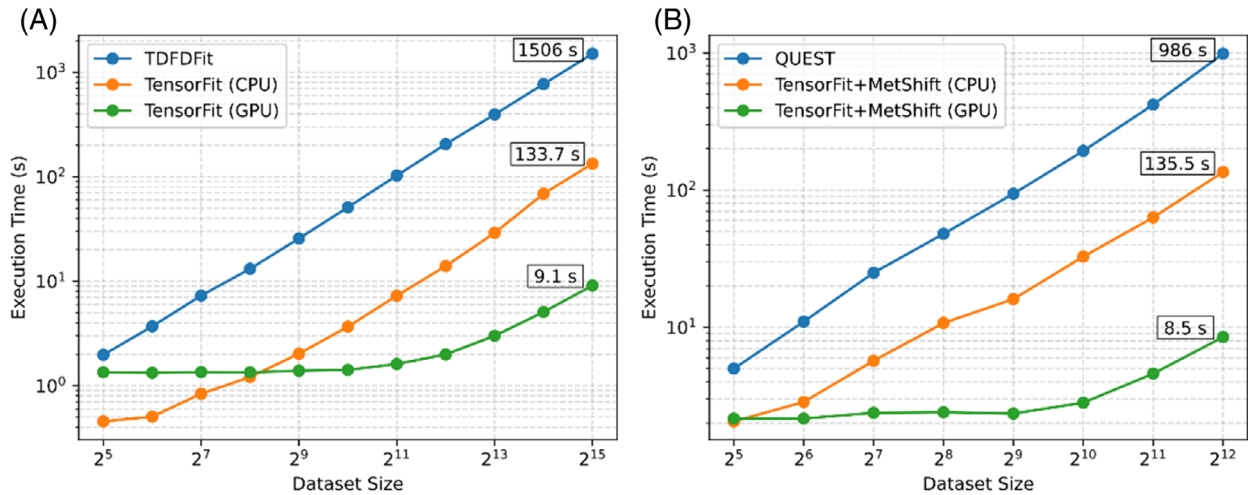


FIGURE 4 Comparison of fitting tool time efficiency. The execution time is presented as a function of the number of spectra. (A) TensorFit (CPU and GPU versions) and TDFDFit. (B) TensorFit + MetShift and QUEST.

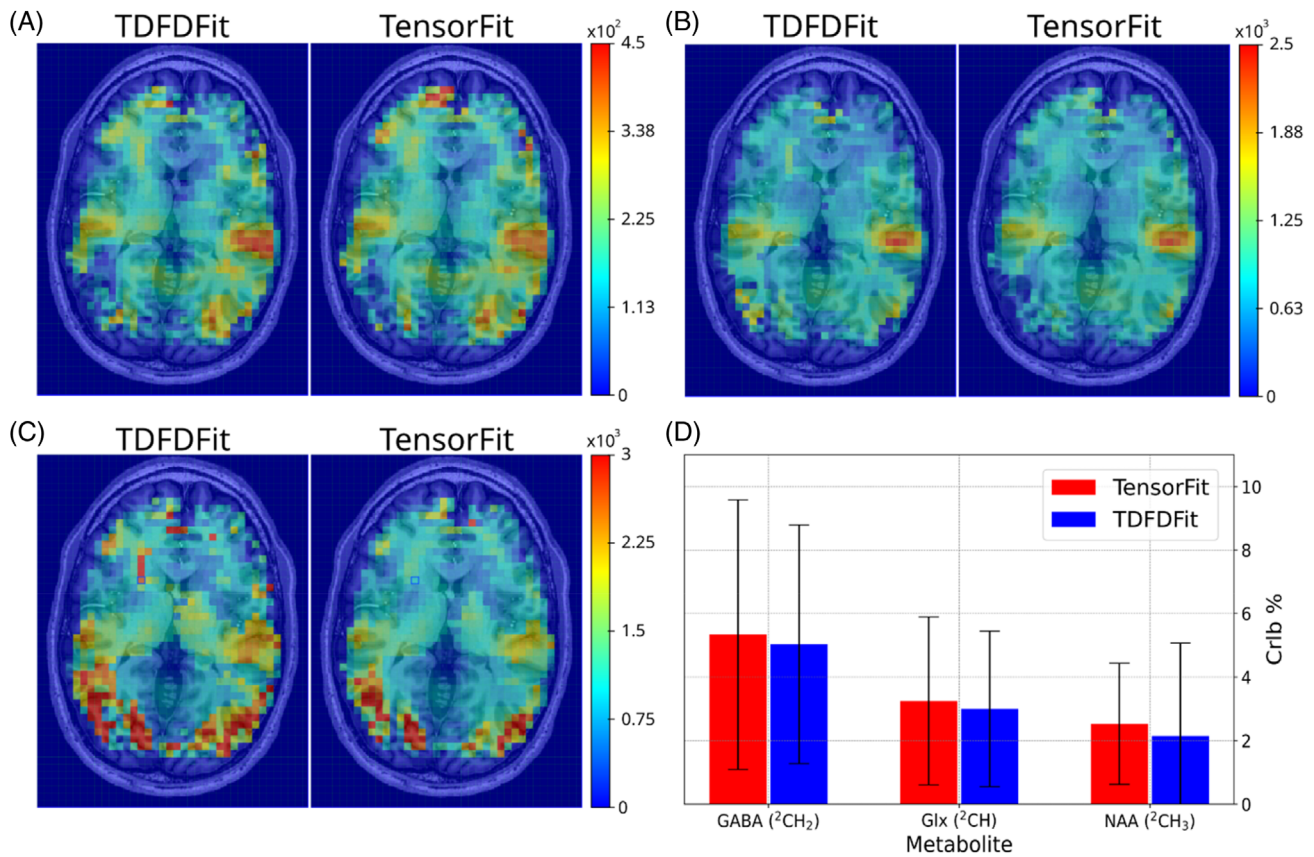
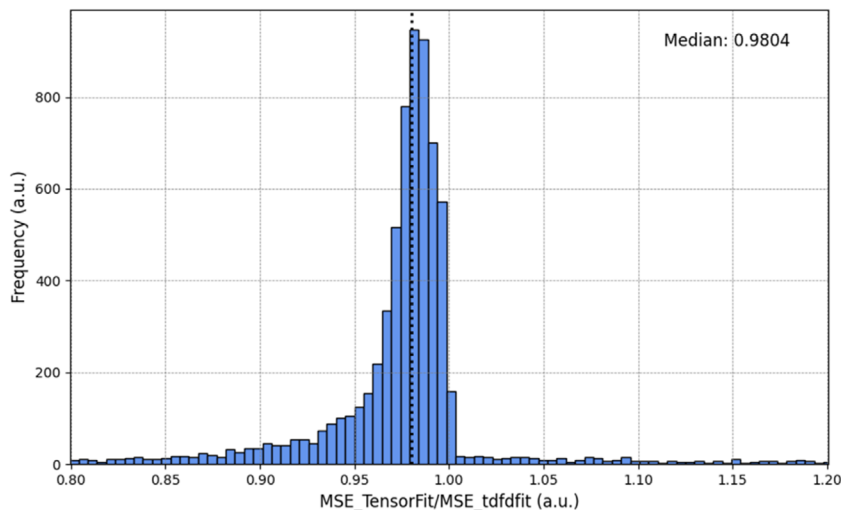


FIGURE 5 Performance comparison of TensorFit and TDFDFit in in vivo data. Metabolites maps for gamma-aminobutyric acid (GABA; A), glutamate + glutamine (Glx; B), and N-acetylaspartate (NAA; C). (D) The averaged Cramér-Rao Lower Band (CRLB) for the three metabolites in the shown slice. The error bars represent the SD. The observed occipital/temporal hotspots are pulse sequence-related and independent of the fitting tool.

FIGURE 6 Histogram distribution of the ratio between the fit-quality number for both methods, computed as the loss of TensorFit divided as the loss of TDFDFit. The vertical line represents the median value.



shows the metabolites maps for three metabolites of interest, in this case, gamma-aminobutyric acid, glutamate + glutamine, and NAA. The columns represent both methods: TDFDFit and TensorFit. Figure 5D shows the CRLB%, illustrating the performance of both tools.

As no ground truth is available for the in vivo data set, we computed the error using the loss function value used for TDFDFit and TensorFit fitting. Figure 6 depicts a histogram illustrating the distribution of voxels with a specific ratio between the loss when using TensorFit and TDFDFit. A ratio smaller than one indicates a better fit using TensorFit. The median value, represented by a vertical line, remains at 0.98, with most (92.4%) cases exhibiting a ratio smaller than one. For this data set consisting of a total of 7098 spectra, the execution time for TDFDFit was 2294 s (38:23 min), whereas TensorFit, running on a GPU, completed the task in 13.5 s, yielding a speed-up factor of 169 \times . When using TensorFit on the CPU, the time extends to 2:40 min, corresponding to a speed-up of 14 \times .

4 | DISCUSSION

In this work, we proposed the off-label use of a DL framework for rapid metabolite fitting of MRS data. Specifically, we used the automatic differentiation capabilities of Torch, using them as a highly parallelized error-minimization algorithm. This approach allowed us to use GPU processing or even obtain a strong acceleration when working solely on the CPU. Focusing on error minimization using traditional DL techniques, the process involved defining an appropriate optimizer for error minimization, implementing the TensorFit tool, and comparing performance with well-established fitting methods.

Many other frameworks, apart from Torch, allow the automatic differentiation of arbitrary computational graphs. TensorFlow,⁴⁸ for example, offers comparable

optimization on GPU and supports complex-valued back-propagation. However, it lacks the flexibility to modify the computational graph during runtime and extends the network initialization time significantly. Other highly optimized frameworks, like JAX,⁴⁹ support automatic differentiation but restrict GPU use to Linux and MacOS platforms. In contrast, Torch allows its implementation in all commonly used operative systems.

First, we investigated the different optimizers available in the Torch library for error minimization. From the comparison, shown in Figure 2, we found that Rprop is the fastest optimizer. Consequently, we adopted Rprop as the optimizer for our TensorFit implementation. Nevertheless, further research should establish whether Rprop maintains its speed advantage across different models and data sets.

We tested TensorFit on simulated data and compared its performance with conventional fitting methods. In terms of fitting accuracy, TensorFit showed a lower percentage error for each fitted metabolite (Figure 3). On average, across all metabolites, the TensorFit error with respect to the ground truth was slightly lower than that of both TDFDFit and QUEST. This can be attributed to the influence of the outliers. In a small number of cases, TDFDFit determines a wrong initial value for the frequency shift, leading to convergence at suboptimal local minima. In contrast, TensorFit, starting from the same values, converges correctly. In addition, we attribute the higher precision of TensorFit with respect to QUEST to its high sensitivity to an initial frequency shift mismatch. For all (tested) fitting algorithms, the percentage error is, on average, smaller than the corresponding CRLB for all components. This behavior is explained by the fact that CRLB represents the minimum variance for an unbiased estimator; this means that most estimations should fall within the range defined by CRLB. Therefore, the average percentage error should be in the same order of magnitude. Further

analysis of the distribution of both metrics is provided in Supporting Material 4.

Regarding time efficiency, when implemented on the CPU, TensorFit outperformed TDFDFit and QUEST by a factor of 7× and 11×, respectively. We attribute the speed-up with respect to TDFDFit to its use of explicit numerical differentiation, which involves several model evaluations. Compared with QUEST, the difference in performance on the CPU stems from the lack of parallelization, whereas TensorFit uses all the computational optimizations of Torch. In addition, as expected, the GPU implementation of TensorFit is up to 165 times faster than TDFDFit and 115 times faster with respect to QUEST. This is due to the high number of processors and the Torch's strong optimization of matrix multiplications.

The performance of TensorFit on the in vivo data set yields better results in terms of mean squared error compared with TDFDFit, strongly suggesting more accurate fitting. We found TensorFit to perform better regarding fit quality in 92.4% of the cases (Figure 6). Although the spatial distribution of metabolites remains the same, there is a slight difference in amplitude and clear outliers, primarily in the case of NAA, in which lipid contamination is significant (Figure 5). A comparison with QUEST was not feasible, as it does not allow the use of Gaussian lineshapes in the model as defined by Eq. (4).

Using Torch for spectral fitting offers several advantages. The main one is the speed-up when executing on a GPU. Additionally, Torch allows high flexibility in defining and customizing fitting models. Nevertheless, it should be noted that some inherent disadvantages of TDFDFit and QUEST, such as high susceptibilities to artifacts, lipid contamination, and problems with overlapping peaks, are also present in TensorFit. This limitation arises from the fact that they all depend on the underlying fitting model. Additionally, the current implementation of TensorFit does not account for the (macromolecular) baseline (i.e., it only performs correctly for long TEs). This will be included in the next release.

In summary, we have introduced a novel tool for metabolite fitting that outperforms existing methods concerning accuracy and time efficiency. The improvement in time efficiency directly addresses a critical obstacle in the clinical application of MRSI, namely the long data-processing times. To illustrate its impact, TensorFit can process a whole brain data set containing 7098 spectra in under 14 s, whereas TDFDFit requires 38 min to analyze the same data set.

5 | CONCLUSIONS

We implemented an ultrafast tool for metabolite fitting of large MRS data sets using Torch's computational

graph auto-differentiation capabilities. We demonstrated the high performance and accuracy of the method with proof-of-principle analysis in simulated and in vivo data sets. We found a superior performance in time efficiency and accuracy, obtaining a speed-up of 165× and 115× when compared with TDFDFit and QUEST, respectively. We believe this work will substantially enhance the use of EPSI sequences or other high-resolution MRS acquisition methods, such as ECCENTRIC, by reducing the required metabolite fitting time.

ACKNOWLEDGMENTS

The project leading to this application has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 813120. Additional funding from the Schweizerischer National fonds zur Förderung der Wissenschaftlichen Forschung, SNSF-182569 and SNSF-207997. Open access funding provided by Inselspital Universitätsspital Bern.

DATA AVAILABILITY STATEMENT

The model and source code that support the findings of this study are openly available at <https://github.com/TuchoTurco/TensorFit-Paper>. The in vivo data used for this study can be obtained upon reasonable request to the corresponding author.

ORCID

Federico Turco  <https://orcid.org/0000-0002-1632-526X>

Milena Capiglioni  <https://orcid.org/0000-0003-0695-0229>

Guodong Weng  <https://orcid.org/0000-0003-3684-5649>

Johannes Slotboom  <https://orcid.org/0000-0001-5121-9852>

REFERENCES

1. Lukas L, Devos A, Suykens JAK, et al. Brain tumor classification based on long echo proton MRS signals. *Artif Intell Med*. 2004;31:73-89.
2. Tate AR, Griffiths JR, Martínez-Pérez I, et al. Towards a method for automated classification of 1H MRS spectra from brain tumours. *NMR Biomed*. 1998;11:177-191.
3. Negendank W. Studies of human tumors by MRS: a review. *NMR Biomed*. 1992;5:303-324.
4. Villanueva-Meyer JE, Mabray MC, Cha S. Current clinical brain tumor imaging. *Neurosurgery*. 2017;81:397-415.
5. Lichy MP, Plathow C, Schulz-Ertner D, Kauczor HU, Schlemmer HP. Follow-up gliomas after radiotherapy: 1H MR spectroscopic imaging for increasing diagnostic accuracy. *Neuroradiology*. 2005;47:826-834. doi:10.1007/s00234-005-1434-0
6. Horská A, Barker PB. Imaging of brain tumors: MR spectroscopy and metabolic imaging. *Neuroimaging Clin*. 2010;20:293-310.
7. Thust SC, van den Bent MJ, Smits M. Pseudoprogression of brain tumors. *J Magn Reson Imaging*. 2018;48:571-589.

8. van der Graaf M. In vivo magnetic resonance spectroscopy: basic methodology and clinical applications. *Eur Biophys J*. 2010;39:527-540.
9. Kreis R. Issues of spectral quality in clinical 1H-magnetic resonance spectroscopy and a gallery of artifacts. *NMR Biomed*. 2004;17:361-381. doi:10.1002/NBM.891
10. Gruetter R, Boesch C. Fast, noniterative shimming of spatially localized signals. In vivo analysis of the magnetic field along axes. *J Magn Reson*. 1992;96:323-334. doi:10.1016/0022-2364(92)90085-L
11. Juchem C, Nixon TW, Diduch P, Rothman DL, Starewicz P, de Graaf RA. Dynamic shimming of the human brain at 7 tesla. *Concepts Magn Reson Part B Magn Reson Eng*. 2010;37B:116-128. doi:10.1002/CMR.B.20169
12. Van Vaals JJ, Bergman AH. Optimization of eddy-current compensation. *J Magn Reson*. 1990;90:52-70. doi:10.1016/0022-2364(90)90365-G
13. Ebel A, Soher BJ, Maudsley AA. Assessment of 3D proton MR echo-planar spectroscopic imaging using automated spectral analysis. *Magn Reson Med*. 2001;46:1072-1078.
14. Ebel A, Maudsley AA. Improved spectral quality for 3D MR spectroscopic imaging using a high spatial resolution acquisition strategy. *Magn Reson Imaging*. 2003;21:113-120.
15. Weng G, Radojewski P, Sheriff S, et al. SLOW: a novel spectral editing method for whole-brain MRSI at ultra high magnetic field. *Magn Reson Med*. 2022;88:53-70.
16. Klauser A, Strasser B, Bogner W, et al. ECCENTRIC: a fast and unrestrained approach for high-resolution in vivo metabolic imaging at ultra-high field MR. *arXiv Prepr arXiv230513822* 2023.
17. Provencher SW. Estimation of metabolite concentrations from localized in vivo proton NMR spectra. *Magn Reson Med*. 1993;30:672-679.
18. Wilson M, Reynolds G, Kauppinen RA, Arvanitis TN, Peet AC. A constrained least-squares approach to the automated quantitation of in vivo 1H magnetic resonance spectroscopy data. *Magn Reson Med*. 2011;65:1-12.
19. Ratiney H, Sdika M, Coenradie Y, Cavassila S, van Ormondt D, Graveron-Demilly D. Time-domain semi-parametric estimation based on a metabolite basis set. *NMR Biomed*. 2005; 18:1-13.
20. Slotboom J, Boesch C, Kreis R. Versatile frequency domain fitting using time domain models and prior knowledge. *Magn Reson Med*. 1998;39:899-911. doi:10.1002/MRM.1910390607
21. Binczyk F, Marczyk M, Polanska J. Mixture model based efficient method for magnetic resonance spectra quantification. *Proceedings of the Bioinformatics and Biomedical Engineering Third International Conference, 2015*. Springer International Publishing; 2015:406-417.
22. Das D, Coello E, Schulte RF, Menze BH. Quantification of metabolites in magnetic resonance spectroscopic imaging using machine learning. *Proceedings of the 20th International Conference of Medical Image Computing and Computer Assisted Intervention*. Springer International Publishing; 2017: 462-470.
23. Hatami N, Sdika M, Ratiney H. Magnetic resonance spectroscopy quantification using deep learning. *Proceedings of the 21st International Conference of Medical Image Computing and Computer Assisted Intervention - MICCAI 2018*. Vol 2018. Springer International Publishing; 2018:467-475.
24. Rizzo R, Dziadosz M, Kyathanahally SP, Shamaei A, Kreis R. Quantification of MR spectra by deep learning in an idealized setting: investigation of forms of input, network architectures, optimization by ensembles of networks, and training bias. *Magn Reson Med*. 2023;89:1707-1727.
25. Gurbani SS, Sheriff S, Maudsley AA, Shim H, Cooper LAD. Incorporation of a spectral model in a convolutional neural network for accelerated spectral fitting. *Magn Reson Med*. 2019;81:3346-3357.
26. Shamaei A, Starcukova J, Starcuk Z Jr. Physics-informed deep learning approach to quantification of human brain metabolites from magnetic resonance spectroscopy data. *Comput Biol Med*. 2023;158:106837.
27. Rizzo R, Dziadosz M, Kyathanahally SP, Reyes M, Kreis R. Reliability of quantification estimates in MR spectroscopy: CNNs vs traditional model fitting. *Proceedings of the 25th International Conference on Medical Image Computing and Computer-Assisted Intervention - MICCAI 2022*. Springer International Publishing; 2022:715-724.
28. Paszke A, Gross S, Massa F, et al. PyTorch: an imperative style, high-performance deep learning library. In: Wallach H, Larochelle H, Beygelzimer A, d'Alché-Buc F, Fox E, Garnett R, eds. *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc; 2019:8024-8035 <http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>
29. Bruckner F, Koraltan S, Abert C, Suess D. Magnum.np—a PyTorch based GPU enhanced finite difference micromagnetic simulation framework for high level development and inverse design. *Sci Rep*. 2023;13:12054.
30. Bosch D, Scheffler K. FastPtx: a versatile toolbox for rapid, joint design of pTx RF and gradient pulses using Pytorch's autodifferentiation. *MAGMA*; 2024;37(1):127-138.
31. Feinman R. Pytorch-Minimize: a Library for Numerical Optimization with Autograd. URL: <https://github.com/rfeinman/pytorch-minimize>. 2021.
32. Cavassila S, Deval S, Huegen C, Van Ormondt D, Graveron-Demilly D. Cramer-Rao bound expressions for parametric estimation of overlapping peaks: influence of prior knowledge. *J Magn Reson*. 2000;143:311-320.
33. Cavassila S, Deval S, Huegen C, Van Ormondt D, Graveron-Demilly D. Cramér-Rao bounds: an evaluation tool for quantitation. *NMR Biomed*. 2001;14:278-283.
34. Chong DGQ, Kreis R, Bolliger CS, Boesch C, Slotboom J. Two-dimensional linear-combination model fitting of magnetic resonance spectra to define the macromolecule baseline using FiTAID, a fitting tool for arrays of interrelated datasets. *Magn Reson Mater Phys Biol Med*. 2011;24:147-164.
35. Borbath T, Murali-Manohar S, Dorst J, Wright AM, Henning A. ProFit-1D—a 1D fitting software and open-source validation data sets. *Magn Reson Med*. 2021;86:2910-2929.
36. de Barros N, McKinley R, Knecht U, Wiest R, Slotboom J. Automatic quality control in clinical 1H MRSI of brain cancer. *NMR Biomed*. 2016;29:563-575.
37. Near J, Harris AD, Juchem C, et al. Preprocessing, analysis and quantification in single-voxel magnetic resonance spectroscopy: experts' consensus recommendations. *NMR Biomed*. 2021;34:e4257.
38. Levenberg K. A method for the solution of certain non-linear problems in least squares. *Q Appl Math*. 1944;2:164-168.

39. Golub GH, Pereyra V. The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate. *SIAM J Numer Anal.* 1973;10:413-432.
40. Hestenes MR, Stiefel E. Methods of conjugate gradients for solving linear systems. *J Res Natl Bur Stand.* 1934;49:409-436.
41. Marjańska M, Deelchand DK, Kreis R, 2016 ISMRM MRS Study Group Fitting Challenge Team. Results and interpretation of a fitting challenge for MR spectroscopy set up by the MRS study group of ISMRM. *Magn Reson Med.* 2022;87:11-32.
42. Riedmiller M, Braun H. A direct adaptive method for faster backpropagation learning: the RPROP algorithm. *Proceedings of the IEEE International Conference on Neural Networks.* IEEE; 1993:586-591.
43. Kingma DP, Ba J. Adam: a method for stochastic optimization. arXiv 1412.6980 2014.
44. Yao Z, Gholami A, Shen S, Mustafa M, Keutzer K, Mahoney M. Adahessian: an adaptive second order optimizer for machine learning. *Proc AAAI Conf Artif Intell.* 2021;35:10665-10673.
45. Liu DC, Nocedal J. On the limited memory BFGS method for large scale optimization. *Math Program.* 1989;45:503-528.
46. Starčuk Z, Starčuková J, Štrbák O, Graveron-Demilly D. Simulation of coupled-spin systems in the steady-state free-precession acquisition mode for fast magnetic resonance (MR) spectroscopic imaging. *Meas Sci Technol.* 2009;20:104033.
47. Naressi A, Couturier C, Castang I, De Beer R, Graveron-Demilly D. Java-based graphical user interface for MRUI, a software package for quantitation of in vivo/medical magnetic resonance spectroscopy signals. *Comput Biol Med.* 2001;31:269-286.
48. Abadi M, Agarwal A, Barham P, et al. Tensorflow: large-scale machine learning on heterogeneous distributed systems. arXiv 1603.04467 2016.
49. Bradbury J, Frostig R, Hawkins P, et al. JAX: composable transformations of Python+ NumPy programs. URL: <http://github.com/google/jax>. 2018.

SUPPORTING INFORMATION

Additional supporting information may be found in the online version of the article at the publisher's website.

FIGURE S1. Precision analysis for all fitting algorithm tools used (blue and red bars correspond to TensorFit and TDFDFit, respectively). The percentual error (%) to the ground truth is displayed for each of the nine metabolites considered in the 3T prior knowledge model. For TDFDFit, the mean error is presented after filtering out outliers.

FIGURE S2. Distribution of percentage error between the fitted parameter and its corresponding ground truth (blue bars) in comparison with the percentual Cramér-Rao Lower Bound (CRLB; red bars) for N-acetylaspartate (NAA). The vertical lines correspond to the mean value of each metric.

TABLE S1. The essential parameters required to reconstruct the fitting model for the simulated data set. Both the initial values for the free parameters and the fixed values from prior knowledge are included.

TABLE S2. The essential parameters required to reconstruct the fitting model for the editing data set. Both the initial values for the free parameters and the fixed values from prior knowledge are included.

How to cite this article: Turco F, Capiglioni M, Weng G, Slotboom J. TensorFit: A torch-based tool for ultrafast metabolite fitting of large MRSI data sets. *Magn Reson Med.* 2024;1-12. doi: 10.1002/mrm.30084